

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Previously Presented) A system for generating a graphical user interface for a script session comprising:

a script having a command line interface; and

a script graphical user interface module, communicatively coupled to said script, for parsing information received from execution of said script to determine an input type command and generating an appropriate graphical input mechanism for said command line interface as a function of said input type command.

2. (Original) The system for generating a graphical user interface for a script session according to Claim 1, further comprising an interface unit for presenting said appropriate input mechanism to a user.

3. (Original) The system for generating a graphical user interface for a script session according to Claim 1, further comprising a shell for executing said script.

4. (Original) The system for generating a graphical user interface for a script session according to Claim 3, further comprising a computing device for implementing said shell and said script graphical user interface.

5. (Original) The system for generating a graphical user interface for a script session according to Claim 3, further comprising a server device for implementing said shell and said script graphical user interface module.

6. (Previously Presented) A method of generating a graphical user interface for a script session comprising:

receiving information from a script having a first command line interface;

parsing said information;

determining if said information comprises an input command; and

generating a first graphical user interface comprising an input prompt for said first command line interface.

7. (Original) The method according to Claim 6, further comprising:

generating a second graphical user interface comprising a prompt for specifying a location of said script; and

invoking execution of said script.

8. (Original) The method according to Claim 6, wherein determining if said information comprises an input command comprises determining if said information comprises a tag.

9. (Original) The method according to Claim 8, further comprises:

determining if said tag identifies a free-text type input command; and

determining if said tag identifies an option type input command.

10. (Original) The method according to Claim 9, wherein generating a first graphical user interface comprises generating said first graphic user interface comprising a first character string and a text box if said tag identifies a free-text type input command.

11. (Original) The method according to Claim 9, further comprising determining a plurality of input values if said tag identifies an option type input command.

12. (Original) The method according to Claim 11, wherein generating a first graphical user interface comprising an input prompt comprises generating said first graphic user interface comprising a second character string and a selection box comprising said plurality of input values if said tag identifies an option type input command.

13. (Original) The method according to Claim 6, further comprising sending said first graphic user interface to a graphics display unit.

14. (Original) The method according to Claim 13, further comprising:
receiving a response from said graphics display unit;
parsing said response to determine a user specified input value; and
sending said specified input value to said script.

15. (Currently Amended) A computing device for generating a graphical user interface for a script session comprising:

a bus;
a memory unit, coupled to said bus, for storing software code; and

a processor, coupled to said bus, for executing said software code, wherein execution of said software code implements:

a script having a command line interface, ~~comprising an input command;~~

a script graphical user interface module, communicatively coupled to said script, for parsing information received from execution of said script to determine an input type command and generating a first Java server page comprising a prompt corresponding to said input type command; and

a web server, communicatively coupled to said script graphical user interface module, for serving said Java server page to a web browser.

16. (Original) The computing device for generating a graphical user interface for a script session according to Claim 15, further comprising a shell for executing said script.

17. (Original) The computing device for generating a graphical user interface for a script session according to Claim 15, wherein said script graphical user interface module generates a second Java server page comprising a prompt for specifying a location of said script.

18. (Original) The computing device for generating a graphical user interface for a script session according to Claim 17, wherein said script graphical user interface module establishes a first communication link between said script graphical user interface module and said script.

19. (Original) The computing device for generating a graphical user interface for a script session according to Claim 18, wherein said script graphical user interface module establishes a second communication link between said script graphical user interface module and said web server.

20. (Previously Presented) The computing device for generating a graphical user interface for a script session according to Claim 19, wherein said script user interface module:

determines if a user has authorization to invoke said script;

invokes execution of said script at said location if said user has authorization to invoke said script; and

denies access to said script if said user does not have authorization to invoke said script.

21. (Original) The computing device for generating a graphical user interface for a script session according to Claim 20, wherein said script user interface module generates an audit trail of said script.

22. (Previously Presented) The computing device for generating a graphical user interface for a script session according to Claim 17, wherein said script graphical user interface module invokes execution of said script at said location.

23. (Previously Presented) A computer-readable medium containing software code which when executed by a computing device cause the computing device to implement a method of generating a graphical user interface for a script session, comprising:

receiving information from a script having a command line interface;

parsing said information from said script;

determining if said information comprises a tag; and

generating a first graphical user interface corresponding to said command line interface comprising an input prompt when said information comprises a tag.

24. (Original) The method according to Claim 23, further comprising:

generating a second graphical user interface comprising a prompt for specifying a location of said script; and

invoking execution of said script.

25. (Original) The method according to Claim 23, further comprises:

determining if said tag identifies a free-text type input command;

determining if said tag identifies an option type input command.

26. (Original) The method according to Claim 25, wherein generating a first graphical user interface comprises generating said first graphic user interface comprising a first character string and a text box if said tag identifies a free-text type input command.

27. (Original) The method according to Claim 25, further comprising determining a plurality of input values if said tag identifies an option type input command.

28. (Original) The method according to Claim 27, wherein generating a first graphical user interface comprises generating said first graphic user interface comprising a second character string and a selection box comprising said plurality of input values if said tag identifies an option type input command.

29. (Original) The method according to Claim 23, further comprising serving first graphic user interface to a web browser.

30. (Original) The method according to Claim 29, further comprising:
receiving a response from said web browser;
parsing said response to determine a user specified input value; and
sending said specified input value to said script.